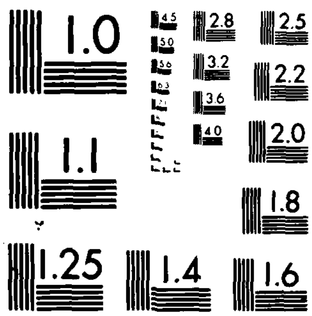
AD-A092 172   GTE PRODUCTS CORP NEEDHAM HEIGHTS MA SYLVANIA SYSTEM--ETC   F/G 17/2
WIDEBAND SPEECH MULTIPLE RATE PROCESSOR STUDY. PART II.(U)
OCT 79   R S CHEUNG, A J GOLDBERG   DCA100-77-C-0054

UNCLASSIFIED   NL

1 OF 1

END
DATE
FILMED
81-2
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL III

12 B.S.

AD A092172

FINAL REPORT

WIDEBAND SPEECH
MULTIPLE RATE PROCESSOR STUDY

DCA 100-77-C-0054

PART II

OCTOBER 1, 1979

GTE SYLVANIA INCORPORATED
SYLVANIA SYSTEMS GROUP
EASTERN DIVISION

12

FINAL REPORT

WIDEBAND SPEECH MULTIPLE RATE PROCESSOR STUDY

DCA 100-77-C-0054

PART II

October 1, 1979

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A092172 (9)

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
FINAL REPORT - A092171
(6) Wideband Speech Multiple Rate Processor Study, Part II.

5. TYPE OF REPORT & PERIOD COVERED
FINAL REPORT.
Sep[redacted] 77 — Sep[redacted] 79

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) R. S. Cheung a[redacted] A. J. Goldberg (15)

8. CONTRACT OR GRANT NUMBER(s)
DCA 100-77-C-0054

9. PERFORMING ORGANIZATION NAME AND ADDRESS
GTE Sylvania Inc.
77 "A" Street
Needham Heights, MA 02194

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Communications Agency
Contract Management Division Code 260 (11)
Washington, D.C. 20305

12. REPORT DATE
1 Oct[redacted]79

13. NUMBER OF PAGES
Part I - 47; Part II - 25

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
(12) 29

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Multiple rate processor, embedded coding schemes, non-embedded coding schemes, split-band voice coding, quadrature mirror filters.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This report presents the results of our investigations on the utility of multiple-rate processing (MRP) terminals in facilitating wideband/narrowband communications. In addition, the design and development of a real-time embedded MRP scheme which transmits speech at the data rates of 2.4, 8.0, 9.6. and 16.0 Kb/s is discussed.

(cont'd)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

406451

20. Abstract (Cont'd)

This report is written in two parts. Part I contains a description of non-embedded and embedded MRP schemes. Detailed treatments on two embedded algorithms, namely, the Linear Predictive Coding/Adaptive Predictive Coding with Adaptive Quantization (APC/APCQ); the Linear Predictive Coding/Split-Band Voice Coding (LPC/SBVC) are included. Part II contains the information on the real-time implementation of the LPC/SBVC coder on the government-owned Sylvania Programmable Signal Processors (PSP). The hardware and programming aspects of the high-speed multiplier-accumulator in the PSP's are also discussed.

TABLE OF CONTENTS

Section		Page
	LIST OF ILLUSTRATIONS	ii
	LIST OF TABLES	iii
IV	4.0 REAL-TIME IMPLEMENTATION	
	4.1 Introduction	4-1
	4.2 The Real-Time MRP Program	4-2
	4.2.1 Interrupt Processing	4-2
	4.2.2 Synchronization	4-3
	4.2.3 The LPC/SBVC Algorithm	4-4
	4.3 MRP Program User's Guide	4-11
Appendix A	The Multiplier-Accumulator Hardware	
A.1	General Description	A-1
A.2	Programming of MULACC	A-3

LIST OF ILLUSTRATIONS

Figure		Page
4-1	Block Diagram of the LPC/SBVC Analyzer and Synthesizer	4-6
4-2	First Stage Quadrature Mirror Filtering of the Error Signal	4-7
4-3	Computations of Pitch Gain and Error Signal	4-8
4-4	Reduced Waveform Computation	4-9
A-1	Block Diagram of the High Speed Multiplier-Accumulator	A-2
A-2	A PSP Program to Demonstrate the Utility of MULACC	A-10
	(Cont'd)	A-11

LIST OF TABLES

Table		Page
A-1	MULACC Instructions	A-4
A-2	Multiplier Functions	A-6
A-3	Instructions to Read Buffer Memories and Multiplier Products	A-7
A-4	Notes on Programming the MULACC	A-8

SECTION IV

REAL-TIME IMPLEMENTATIONS

4.1 Introduction

In the DCA Wideband Speech Multiple-Rate Study (DCA 100-77-C-0054), GTE Sylvania investigated the applications of embedded mutliple-rate processing (MRP) terminals in facilitating wideband/narrowband communications. To simulate the performance of these terminals, a real-time implementation of the MRP algorithm, Linear Predictive Coder/Split-Band Voice Coder (LPC/SBVC), was developed for the government-owned Sylvania Programmable Signal Processors (PSP) with special multiply-accumulate hardware. Part II of this final report briefly discusses the real-time program and provides information on special programming techniques needed to program the multiply-accumulate hardware.

Operations of the LPC/SBVC algorithm, as discussed in Part I of this report, include the linear prediction analysis, computation of the residual signal, splitting of the frequency band of the LPC error signal, and quantization of the subband waveforms. Though the above embedded MRP scheme appears to be straightforward, its processing requirement is equivalent to a combination of two coders, namely, LPC and SBVC. Henceforth, real-time implementation of the algorithm is generally unthinkable on many machines. In particular, the LPC/SBVC transmitter functions include a LPC analyzer, computation of the LPC residual, three stages of split band filtering, and individual quantization of the eight subbands. Fortunately, the Sylvania PSPs, after modification under the Subband Coder Study (DCA 100-79-C-0001), are equipped with high speed multiplier-accumulators which can multiply two 16-bit numbers and accumulate the 32-bit product with 35-bit precision in 208 nsec. Moreover, this hardware is especially efficient for linear filtering operations. Appendix A gives a detailed

description of the high-speed multiplier-accumulator. With these PSPs, real-time implementation of the LPC/SBVC in half-duplex mode has been possible.

4.2 The Real-Time MRP Program

The real-time LPC/SBVC program, written in PSP assembly language, enables the two signal processors to collect, analyze, transmit, receive, and synthesize speech data in a half-duplex mode. Operations of the software include the initialization of system parameters, interrupt processing, acquisition of synchronization, and the processing of speech via the LPC/SBVC algorithm.

4.2.1 Interrupt Processing

The key to correctly sequencing the operations is the use of two interrupts in the PSP: the speech side interrupt and the line side interrupt. The speech side interrupt is set to interrupt the computer at regular intervals and transfer control to software (starting at location 1). After a frame of 180 speech samples are collected, a Speech Data Ready Flag is set which indicates the beginning of the analyzer. On the other hand, the line side interrupt traps to the service routine which starts at location Ø. It outputs a five-volt level which represents the modem clock and one bit of the transmission data. At every eighth interrupt, eight bits of data are entered. When a frame of 360 bits has been received, an indicator (Data Buffer Ready Flag) is set which signifies the start of the synthesizer.

At the beginning, the program goes to an initialization routine where all parameters and counters of analyzer and synthesizer filters are cleared. It then goes into a loop which waits until one of the two flags (speech

buffer ready flag and receiver ready flag) is set and then it performs the corresponding processing. At the end of the operation, the program returns to the wait loop to use up the idle time left over. Reinitialization can always be achieved by setting bit 15 of the front panel switches.

4.2.2 Synchronization

Due to the complexity of the LPC/SBVC algorithm, the total processing time needed for both transmitter and receiver functions far exceeds the designated frame time of 22.5 msec. As a result, only a half-duplex implementation of the algorithm is possible on the modified PSPs. In other words, the users can choose, via the push-to-talk (PTT) switch on the handset, either the transmitter or the receiver function. If the transmitter function is selected, the sampling of input speech, analysis of the data via the LPC/SBVC algorithm, and the transmission of the 360 binary bits through the MIL-188C or the RS232 digital interface are performed. On the other hand, operations of the receiver call for the decoding of the binary bits into speech parameters, the reconstruction of the speech waveform via one of the four (2.4, 8.0, 9.6, and 16.0 Kb/s) synthesizers, and the outputting of samples through the D/A converter. In order for the receiver to decode the proper information, synchronization has to be established and constantly maintained between the transmitting PSP and the receiving one.

Synchronization between the PSPs is achieved through the use of fixed data patterns. After initialization, the machines will begin in the sync search mode where a (111111100....0) and a (011111100....0) data patterns are transmitted every other frame. The alternate 1 and 0 in the first bit position indicates the beginning of each frame. The following 6 bits

(111111), ordinarily representing the quantized pitch parameter, are employed to signify the sync search mode avoiding the synthesis of meaningless data. Then the receiver builds a histogram based on the frequency of occurrence of 1's within the two frames. Taking the absolute difference of the ith and the $(i + 360)^{th}$ histogram values, the position that has the maximum difference corresponds to the sync point. After determining the sync value, the program automatically goes to the receiver mode, but synchronization between machines will still be maintained at all times. The machine will always stay in the received mode until the push-to-talk switch on the telephone has been pushed. At such time, half-duplex communications can commence between the two PSPs.

4.2.3 The LPC/SBVC Algorithm

After establishing synchronization, the PSPs process speech signals via the LPC/SBVC algorithm. Depending on the position of the PTT switch on the handset, either the analyzer or the synthesizer functions are performed. The block diagram of the LPC/SBVC scheme is shown in Figure 4-1. As illustrated in the figure, the LPC/SBVC analyzer always generate a 16 Kb/s binary bit stream whereas the synthesizer can reconstruct the incoming signal at data rates ranging from 2.4 to 16.0 Kb/s.

The nucleus of the LPC/SBVC software is the LPC-10 version 23* developed for NSA under Contract No. MDA 904-76-C-0378. This version of the LPC program employs Atal's covariance approach to perform the tenth order linear prediction analysis. Reflection coefficients are computed through Cholesky decomposition of the covariance matrix. Pitch is obtained using the Average Magnitude Difference Function (AMDF). Synthesis of the LPC waveform are performed with the tenth order recursive filter. Operations

of the LPC-10 program are well documented in the Final Report delivered under the Contract, and they will not be repeated here. In particular, the bordered boxes shown in Figure 4-1 represent those that have been covered in the Final Report [1].

In addition to the LPC-10 functions, operations of the LPC/SBVC analyzer also include the generation of the reduced waveform, computation of the pitch gain, the error signal, and quadrature mirror filtering with 3 stages of 12-tap filters. The reduced waveform is computed in this algorithm by subtracting the incoming signal with one estimated from the short-term prediction loop as shown in Figure 4-2. Initially, the reflection coefficients obtained as a result of the linear prediction analysis are converted to predictor coefficients. To minimize the computational time, only a fourth order predictor is utilized to generate the reduced waveform. Then long-term prediction employing a first order pitch loop is applied. To begin, a pitch gain parameter is calculated in the manner as depicted in Figure 4-3. Also illustrated in the figure, the error signal is formed by subtracting samples of the reduced waveform from the pitch predicted one. Then three stages of quadrature mirror filters are exploited to split the frequency band of the error signal to eight subbands. The first filtering stage using the multiplier-accumulator (MULACC) hardware is shown in Figure 4-4. As depicted in the flowchart, the memories of MULACC have to be preloaded. Particularly, the X-buffer is loaded with low-band filter $h_1(n)$ followed by high-band filter $h_2(n)$ coefficients, and the Y-buffer is loaded with the error signal together with 12 samples of its previous history. Setting the X-buffer pointer to be at the first coefficient of the low band filter, and initializing the Y-buffer pointer to be at the error signal sample $E(j)$, low pass

[1] GTE Sylvania Inc., "Final Report for the LPC-10 Feasibility Study," Contract No. MDA 904-76-C-0378, January 1977.

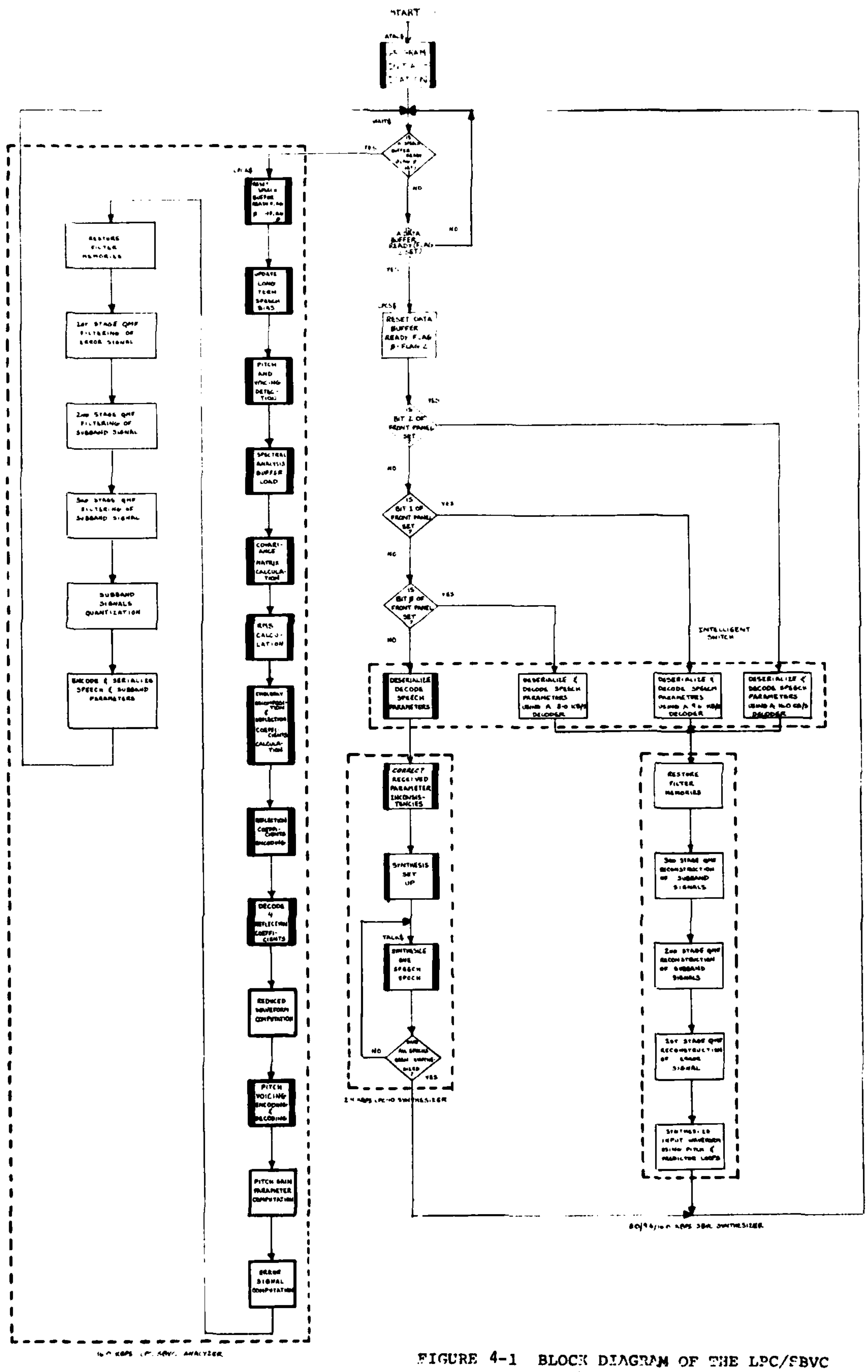


FIGURE 4-1 BLOCK DIAGRAM OF THE LPC/SBVC ANALYZER AND SYNTHESIZER

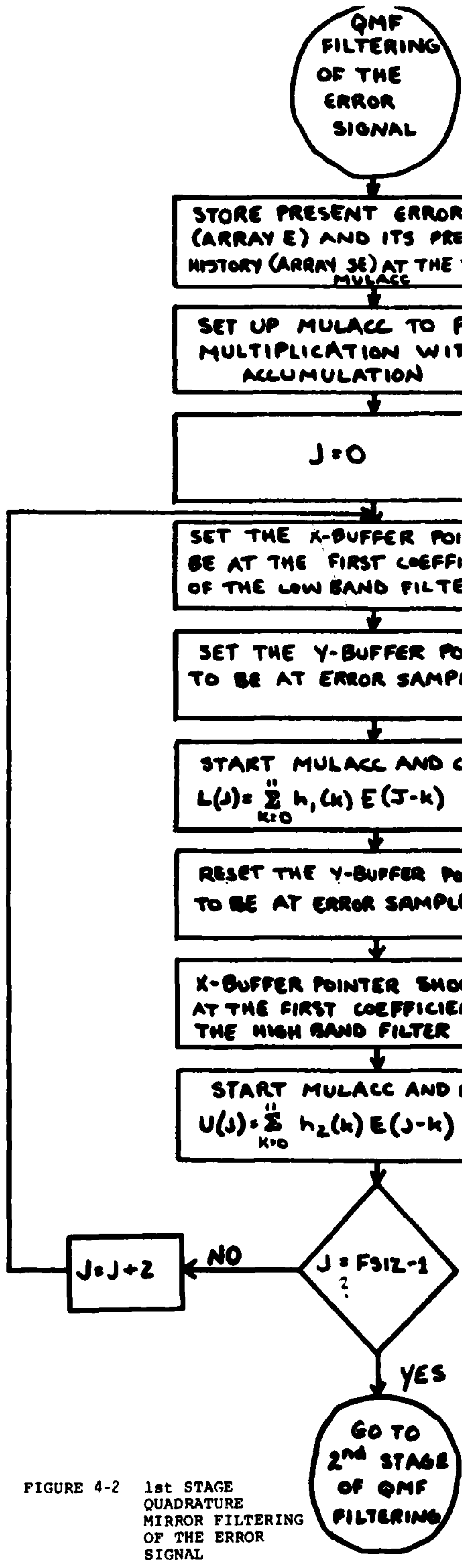


FIGURE 4-2 1st STAGE QUADRATURE MIRROR FILTERING OF THE ERROR SIGNAL

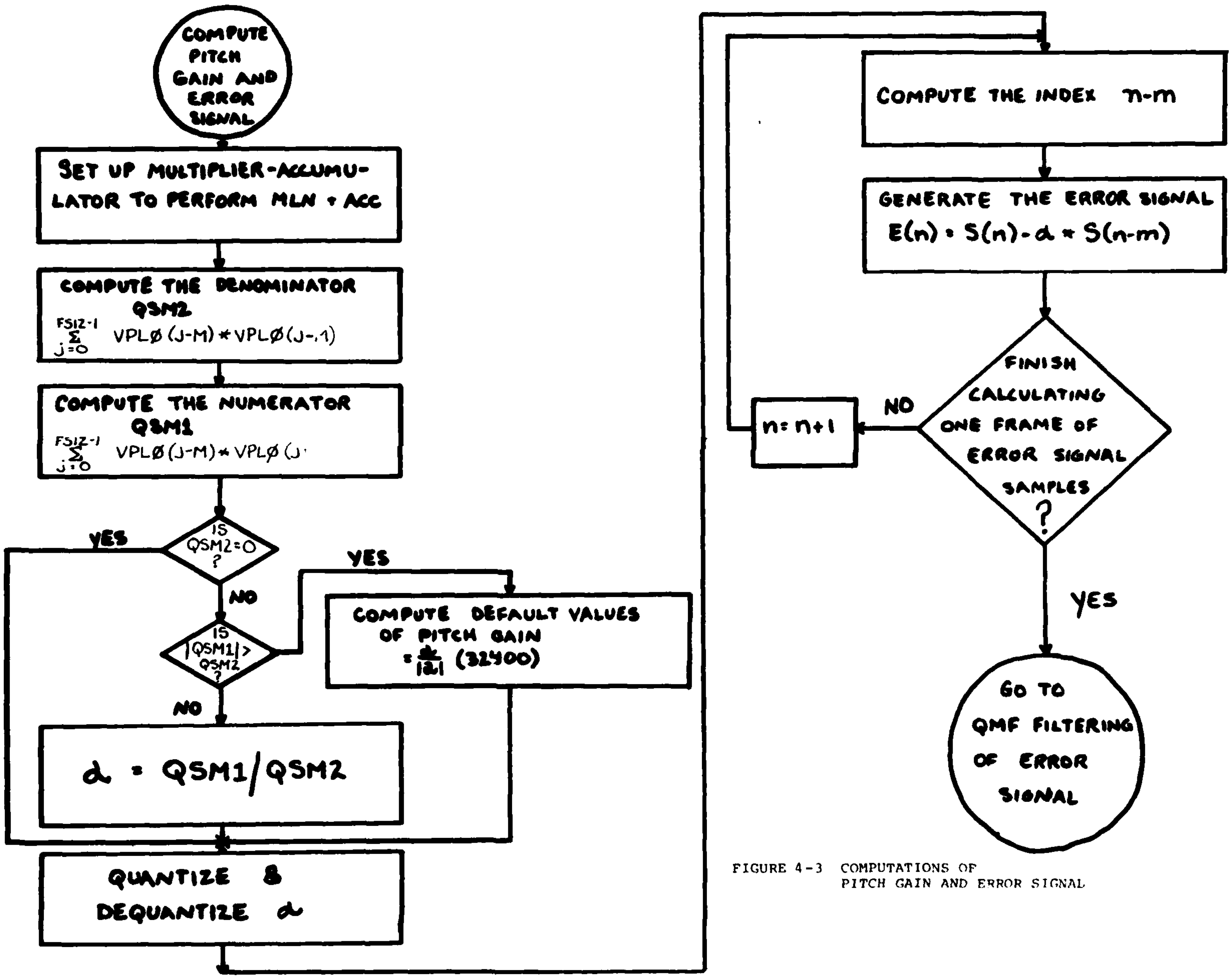


FIGURE 4-3 COMPUTATIONS OF PITCH GAIN AND ERROR SIGNAL

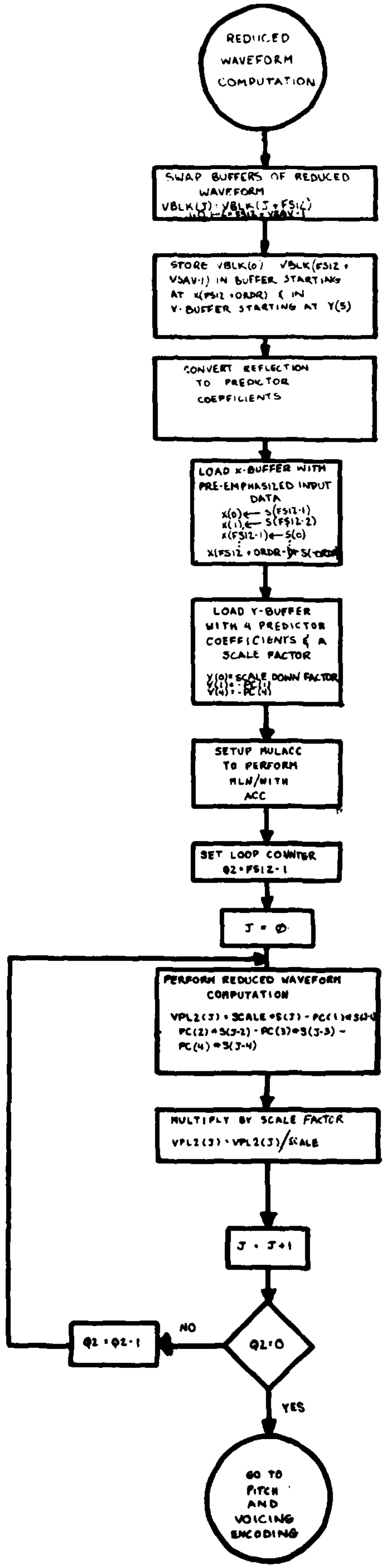


FIGURE 4-4 REDUCED WAVEFORM COMPUTATION

filtering of the j^{th} sample is performed. Resetting the Y-buffer pointer to be at E(j) again, high pass filtering of the signal is done. Similarly, the second and third stages of quadrature mirror filtering are accomplished with MULAC. Then the seven out of the eight subband signals are quantized for transmission.

At the receiver, the binary bit streams are stored in a double-buffer manner. Based on the acquired sync value, the correct 360 bits for the frame are obtained. Then the parameters are decoded according to the front panel switches of the PSP. With a switch setting of Ø (i.e., all switches are up), 54 bits out of the 360 are decoded into speech parameters for 2.4 Kb/s LPC-10 synthesis. If the value is 1 (i.e., all switches are up except switch Ø), 180 bits/frame are decoded for 8.0 Kb/s SBVC synthesis. With a switch setting of 2 (i.e., all swtiches are up except switch 1), 216 bits are peeled off from the 360 bits for 9.6 Kb/s SBVC synthesis. If the value is 4 (i.e., all switches are up except 2), the entire 360 bits are employed for the 16 Kb/s SBVC synthesis.

4.3 MRP Program User's Guide

I. LOADING

Load the program MRP.PBJ into both PSP's using procedures as described in the PSP Loader Program PSPLD.

II. OPERATING PROCEDURES

After the program has been loaded, depress bit 15 of PSP front panel switches. Then start the MRP program by setting the RUN/STOP switch to RUN. In this mode, the program will loop around an initialization routine which clears out data buffers and resets loop registers. When bit 15 is placed in the up position, the software will first attempt synchronization between the two PSP's and then proceed to process speech using the MRP algorithm. If re-synchronization is desired, hold bit 15 of both PSP's in the down position and return it to the up position. Since the MRP program only functions in the half-duplex mode, the push-to-talk switch in the handset has to be pushed to complete synchronization and establish speech transmission. Options of the MRP coder program are summarized as:

CONSOLE SWITCH REGISTER

bit 15	DOWN:	INITIALIZATION
	UP:	NORMAL OPERATION
bit 2	DOWN:	SELECTS 16 KB/S RECEIVER
bit 1	DOWN:	SELECTS 9.6 KB/S RECEIVER
bit Ø	DOWN:	SELECTS 8.0 KB/S RECEIVER
	UP:	SELECTS 2.4 KB/S LPC-10

Appendix A

The Multiplier-Accumulator Hardware

A.1 General Description

GTE Sylvania developed a state-of-the-art multiplier-accumulator (MULACC) under contract DCA 100-79-C-0001 which greatly enhances the capability of the Sylvania Programmable Signal Processor (PSP) in various signal processing applications. In addition to linear convolutions, the hardware performs computations of auto-/cross-correlation of two arrays of numbers most effectively. The nucleus of the design is a high speed 64 pin multiplier-accumulator chip (TRW TDC 1010J) which multiplies two 16-bit numbers and accumulates a 35-bit product in 115 nsec. Moreover, to fully exploit its capability, two buffers of fast RAM memories (MOSTEK MK4118P) are included which feed data directly into the chip. Since the circuitry communicates with the PSP CPU through input/output buses, these memory buffers drastically reduce the passing of data between the PSP and MULACC. Furthermore, since MULACC is treated as a peripheral, no hardware changes are required on the PSP CPU and all existing programs are still operable on the modified machine. After starting MULACC, the PSP CPU is also free to perform other tasks and this represents a more efficient utilization of available processing time. As an indication of its speed, the MULACC is capable of accessing the data from the two buffers, multiplying two 16-bit numbers, and forming a 35-bit product accumulation in 208 nsec (or 2 PSP cycles).

The MULACC hardware as shown in Figure A-1 consists of two PC cards. The first card (board CON) , located at slot 7 of the PSP nest, is responsible for interpreting outputs from the PSP CPU, and decoding them into MULACC instructions. In particular, outputs from channel 5 of the PSP are converted into MULACC instructions whereas outputs from channel 6 are treated as data. Then the correct calling sequence for programming MULACC becomes

1) OUT 5 INSTRUCTION
2) OUT 6 DATA

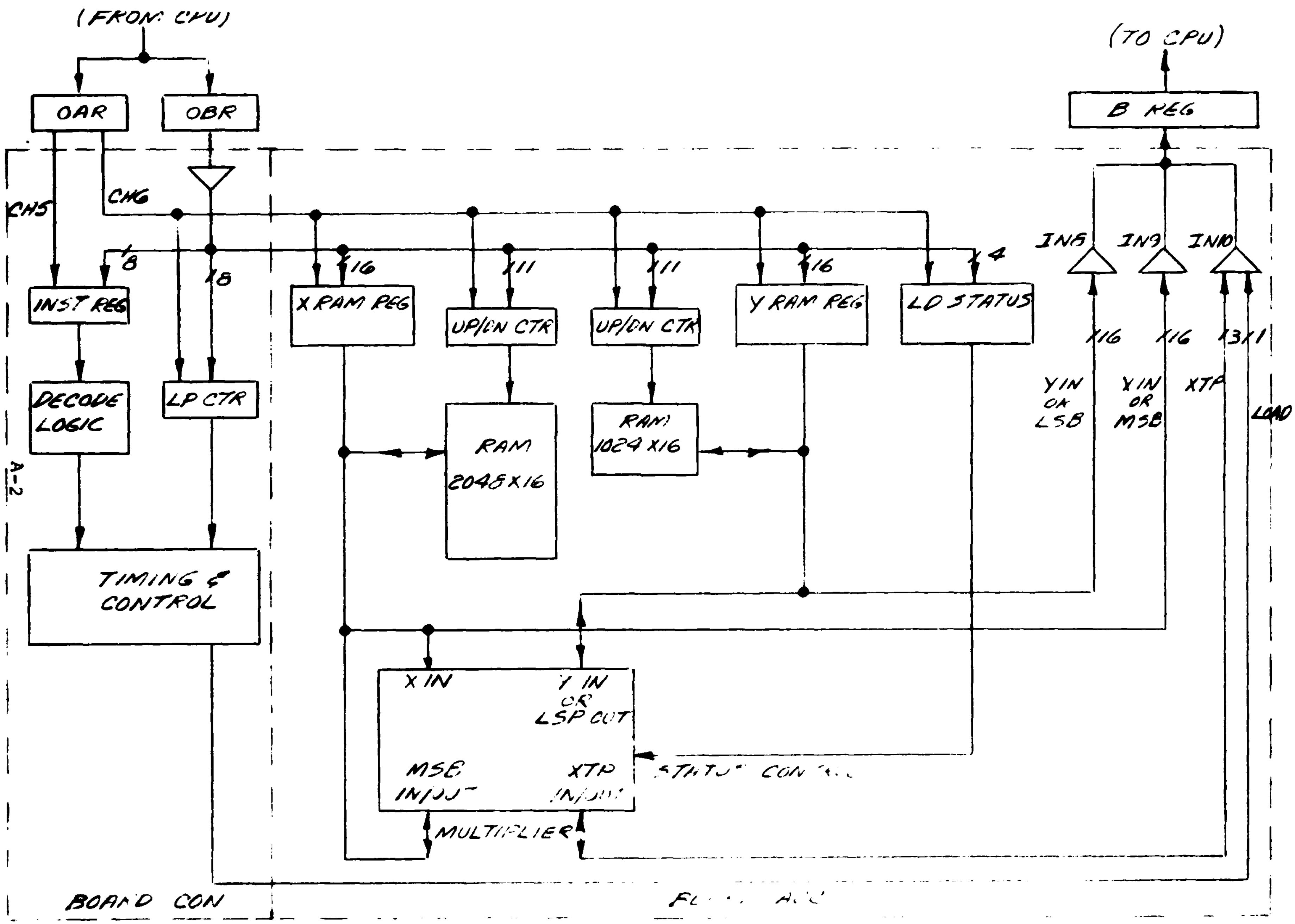


FIGURE A-1 BLOCK DIAGRAM OF THE HIGH SPEED MULTIPLIER-ACCUMULATOR

After the first line of code, the instruction register of MULACC is loaded and the corresponding timing and control information are read from a programmable logic array (PLA). This is later used to set up the sequence of operations required by the instruction. The second program instruction provides operands for the corresponding memory registers and loopcounters.

The second PC card (board ACC) located at slot 5 of the PSP nest, contains the two buffer memories as denoted by X and Y, and the multiplier-accumulator chip. Memory address registers which are also present on board ACC can be programmed to point to any locations within the buffers. An up/down counter is utilized to increment/decrement the address registers automatically after each memory access. In the present design, 2K 16-bit data points can be pre-stored in X-buffer whereas 1K can be loaded into Y-buffer. After MULACC is started with setting of the loop counter, a 32-bit product is first formed which is then added to the content of the TRW chip's output register. Upon the termination of the operation, a LOAD bit is set to a 1 and the 35-bit product can be transferred to the PSP input channels in three segments; namely, extended, most significant, and least significant products.

A.2 Programming of MULACC

As discussed in Section A.1, procedures for programming MULACC include the outputting of an instruction code from channel 5 of the PSP, followed by an output of data from channel 6. From the first output, the instruction register of MULACC is loaded and the second output supplies the operand data. MULACC instructions, which include the loading of X and/or Y buffer addresses and memories, the selection of multiplier functions, and loading of multiplier loop counter are listed in Table A-1.

As for multiplier functions, the hardware permits the choice of 12 functions; namely, unsigned magnitude/2's complement arithmetic, multiplication with/without rounding, multiplication with/without accumulation, multiplication with/without subtraction, increment/decrement of X-buffer pointers, and increment/decrement of Y-buffer pointers. Bit settings of the corresponding functions

TABLE A-1 MULACC INSTRUCTIONS

INSTRUCTIONS	HEX VALUES	COMMENTS
Idle (IDLE)	0	An output of value 0 to channel 5 will set the MULACC to an idle state
Load X-Buffer Address (XADR)	1	Outputs of value 1 to channel 5, followed by the number XXXX to channel 6 will point to address XXXX of X-Buffer ($0 \leq$ XXXX $\leq$ 2047).
Load Y-Buffer Address (YADR)	2	Outputs of value 2 to channel 5 followed by the number YYYY to channel 6 will point to address YYYY of Y-Buffer ($0 \leq$ YYYY $\leq$ 1023).
Load Identical X-Buffer and Y-Buffer addresses (ADR)	3	Outputs of value 3 to channel 5, followed by the number XXXX to channel 6 will set both X and Y-Buffer address pointers to XXXX.
Load X-Buffer Memory (LDX)	4	Outputs of value 4 to channel 5, followed by the number DDDD to channel 6 will load data DDDD into the pointing X-Buffer address (see Note 1 in Table A-4).
Load Y-Buffer Memory (LDY)	5	Outputs of value 5 to channel 5, followed by the number DDDD to channel 6 will load data DDDD into the pointing Y-Buffer address (see Note 1 in Table A-4).
Load Identical Values into X and Y-Buffer Memories (LXAY)	6	Outputs of value 6 to channel 5, followed by the value DDDD to channel 6 will load data DDDD into the pointing X and Y-Buffer addresses (see Note 1 in Table A-4).
Select Functions (TASK)	7	Outputs of value 7 to channel 5 followed by the value DDDD to channel 6 set up the MULACC to perform task DDDD. (Refer to table A-2 for specific task definitions.)
Read Multiplier (READ)	8	An Output of value 8 to channel 5 followed by a waiting period of 4 cycles will enable the products presently residing at the MULACC to be read (see Note 2 and refer to Table A-3 for specific reads).
Load Loop Counter & Start Multiplier (LOOP)	9	Ouputs of value 9 to channel 5 followed by the number (255-nnn) to channel 6 will set up the MULACC to multiply nnn times. The maximum number allowed is 255. Immediately after the setting of the loop counter, multiplication begins.
Load different addresses of X and Y-Buffers (XYADR)	A	This is a short cut to load different X and Y-Buffer addresses. Outputs of value A to channel 5 followed by a value of XXXX to channel 6 and a value of YYYY to channel 6 will set the X and Y address pointers to XXXX and YYYY, respectively.

are summarized in Table A-2. To illustrate this, the selection of multiplication with accumulation function is considered. First, a value 7 is outputted from PSP channel 5 to MULACC which signifies the choice of functions. Then an output of $(33)_8$ from channel 6 sets up MULACC to perform 2's complement arithmetic, multiply (with no rounding) with accumulation, and increment both X and Y buffer pointers after each operation. Hence, by resetting the buffer pointers, the multiplier can be started to perform multiplications with accumulation of two arrays of numbers.

When the multiplier is finished, the load (RDY) bit of MULACC will be set to 1 and the results can be transferred. Since the PSP is a 16-bit machine, the 35-bit product is shipped via three input channels; namely, the extended product (XTP) through channel 10, the most significant product (MSP) through channel 9, and the least significant product (LSP) through channel 8. In addition, the RDY bit is multiplexed with the actual XTP during the transfer. So, after reading the XTP, the PSP has to perform a masking followed by a shift right once operation in order to obtain the correct XTP.

Besides the products, input channels 8 and 9 are also utilized to read the X and Y buffer memories. A tri-state device gating on the multiplier RDY bit is used to switch the outputs from the multiplier to that of the memories. In particular, when RDY = 0, input channels 8, 9 are connected to that of Y, X buffers, respectively. On the other hand, if RDY = 1, input channels 8, 9, 10 are hooked up to LSP, MSP, XTP. PSP instructions required to read products and memories are shown in Table A-3. Also, special considerations on loading and reading of MULACC buffers, together with multiplier functions are discussed in Table A-4.

To further illustrate the utility of MULACC, a PSP demonstration program that computes the following operation is detailed in Figure A-2:

$$y(0) = \sum_{k=0}^{199} h(k)x(0-k) \qquad \text{(A-1)}$$

TABLE A-2 MULTIPLIER FUNCTIONS

	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
BIT SETTING = 0	DY	DX	NO RND	NO SUB	NO ACC	USM
BIT SETTING = 1	IY	IX	RND	SUB	ACC	TC

Symbol Definitions:

USM: Unsigned magnitude arithmetic (16-bit with no sign bit)

TC : 2's complement arithmetic (15-bit with 1 sign bit)

ACC: product accumulation (ACC = Product + ACC)

SUB: product subtraction (If bit 2=1 & bit 1=1, ACC = PRODUCT - ACC)

IX : increment X-buffer pointer

DX : decrement X-buffer pointer

IY : increment Y-buffer pointer

DY : decrement Y-buffer pointer

RND: rounding (see Note 3 on Table A-4)

TABLE A-3 INSTRUCTIONS TO READ BUFFER MEMORIES AND MULTIPLIER PRODUCTS

INSTRUCTIONS	PSP MNEMONICS	COMMENTS
To read contents of X-Buffer (XREAD)	INPA 9.	After setting up the X-Buffer address using XADR, its content can be read after a waiting period of 4 cycles
To read contents of Y-Buffer (YREAD)	INPA 8.	After setting up the Y-Buffer address using YADR, its content can be read after a waiting period of 4 cycles
To read the content of extended product (XTPR)	INPA 10.	Following either a load loop counter (LOOP) or read multiplier (READ) instruction, the extended product (XTP) can be accessed. Only the lower four bits are of interest and the upper 12 bits should be masked out. Its format is as follows: 15 ••• 3 2 1 0 ←All 1's→ ← XTP → ←RDY→ where RDY=0: multiplication has not been completed RDY=1: multiplication has been completed The XTP is meaningful only if RDY=1
To read the content of the most significant product (MSPR)	INPA 9.	Following either a load loop counter (LOOP) or a read multiplier (READ) instruction, the most significant product (MSP) can be read. However the MSP value is valid only after the multiplication has been completed (i.e., RDY=1)
To read the content of the least significant product (LSPR)	INPA 8.	Following either a load loop counter (LOOP) or a read multiplier (READ) instruction, the least significant product (LSP) can be read. The LSP value is valid only after the multiplication has been completed (i.e., RDY=1)

TABLE A-4 NOTES ON PROGRAMMING THE MULACC

1. Loading of X and/or Y-Buffer Memories

Since the loading of buffer memories takes more than two cycles, a 4-cycle wait period is needed to assure the success of writing consecutive memory locations.

2. Reading of Buffer Memories and Multiplier Products

The MULACC functions include the readings of buffer memories and the three multiplier products. In executing instructions that do not involve the multiplier, the product outputs are disenabled from the tri-state control and the bus is connected to the buffer memories which allows the reading of their contents. At this time, the multiplier ready bit is reset (RDY = 0). However, by outputting a read multiplier (READ) instruction on channel 5, the tri-state bus is switched to the multiplier outputs. This set the RDY bit to 1 and enables the reading of existing multiplier products.

In performing operations that require the multiplier, the RDY bit, originally set to 0, will change to a 1 immediately upon their completion. At this time, the three products will be ready to be transferred to the PSP using instructions shown in Table A-2.

3. Multiply with Rounding

The TRW chip has a multiply with or with no rounding option. The multiply with rounding is performed by adding a 1 to bit 15 of the least significant product and the rounded result is obtained by reading the extended and most significant products. This option is generally not recommended for multiplication together with accumulation since it yields erroneous results. To further illustrate this, the multiplication with rounding and accumulation of two arrays of 0's shows a non-zero final value.

Block 1 of the program chooses the multiplier functions whereas Block 2 indicates the loading of the X and Y buffer memories. As shown in the figure, the starting addresses of the buffers are first set and then each MULACC buffer location can be individually loaded from PSP memories. By resetting the buffer pointers (X pointing to h(0), Y pointing to x(0)), the loop counter is initialized to be 200 and multiplication with accumulation is started as shown in Block 3. The multiplier RDY bit is constantly checked and results are transferred to PSP as illustrated in Block 4.

```
PAP-11 V31
                               NL=1,C
                                      .TITLE DEMO.PAP
                                      'THIS PROGRAM DEMONSTRATES THE USE OF MULACC TO PERFORM
                                      'BANDPASS FILTERING
                                      '
                                      '  Y(0)=H(0)X(0)+H(1)X(-1)+...+H(199)X(-199)
                                      '
                                      '
                                      '
                                      'REGISTER FILE DEFINITION
                000000001             R1=1
                                      '
                                      '
                                      'COMMAND DEFINTIONS
                00000B500             SETUP=132400                          'SAME OP CODE AS OUTM 5,0000
                000000660             DO=3140                               'SAME OP CODE AS OUTA 6
                000000820             LSPR=4040                             'SAME OP CODE AS INPA 8.  'READ THE LEAST SIG. PART OF PROD
                000000920             MSPR=4440                             'SAME OP CODE AS INPA 9.  'READ THE MOST SIG. PART OF PRODU
                000000A20             XTPR=5040                             'SAME OP CODE AS INPA 10. 'READ THE EXTENDED PART OF PRODUCT
                                      '
                                      '
                                      'MULACC FUNCTION DEFINITIONS
                000000001             XLN=01                                'MULTIPLY WITH NO ROUNDING
                000000002             ACC=02                                'ACCUMULATE
                000000010             IX=20                                 'INCREMENT X BUFFER POINTER
                000000020             IY=40                                 'INCREMENT Y BUFFER POINTER
                                      '
                                      '
                000000008             .PROG 10
                                      '
      0008      B500 0005        1    .INST          SETUP,   TASK          'SELECT MULACC FUNCTION
      0009      7A43 0033                            LDAL     XLN!ACC!IX!IY 'FUNCTIONS ARE MULTIPLY WITH ACC,INCR. X AND INCR. Y
      000A      0660 0000             .INST          DO
                                      '
                                      'LOADING OF BUFFERS X AND Y
      000B      B500 0002             .INST          SETUP,   ADR           'SET UP TO LOAD BOTH ADDRESSES OF BUFX AND BUFY
      000C      7A43 0000                            LDAL     0             'STARTING AT 0
      000D      0660 0000             .INST          DO
                                      '
                                      '
                                      'LOAD FILTER COEFFICIENTS INTO X BUFFER
      000E      3A41 00C7        2                   LDRL     200.-1,R1     'SET UP COUNTER TO LOAD 200 COEFS
      000F      B500 0003             .INST          SETUP,   LDX           'SET UP INSTRUCTION TO LOAD X MEMORY
      0010      CA21 0185     XLOAD$                 LDA      H ,R1         'PICK UP FILTER COEFS
      0011      0660 0000             .INST          DO                     'AND STORE IN BUFX
      0012      1141 0010                            JDI      XLOAD,R1      'LOOP UNTIL ALL 200 COEFS ARE FINISHED
                                      '
                                      '
                                      'LOAD INPUT SAMPLES INTO Y BUFFER
      0013      3A41 018E                            LDRL     200.+199.-1,R1 'SET UP COUNTER TO LOAD 399 SAMPLES IN BUFFER Y
```

FIGURE A-2 A PSP PROGRAM TO DEMONSTRATE THE UTILITY OF MULACC

```
PAP-11 V31                         DEMO.PAP
 0014  B500 0004          .INST        SETUP,   LDY            'SET UP TO LOAD Y BUFFER
 0015  CA21 000A  YLOAD$               LDA      MEMR,R1        'WITH MEMORY CONTENT FROM INPUT BUFFER
 0016  0660 0000          .INST        DO
 0017  1141 0015                       JDI      YLOAD,R1       'LOOP UNTIL ALL 399 SAMPLES ARE DONE
                          ;
                          ;
                          'MULTIP_Y AND ACCUMULATE WITH MULACC
 0018  B500 0000          .INST        SETUP,   XADR           'RESET THE X BUFFER POINTER TO BE AT 0
 0019  7A43 0000                       LDAL     0
 001A  0660 0000          .INST        DO
 001B  B500 0001          .INST        SETUP,   YADR           'RESET THE Y BUFFER POINTER TO BE POINTING
 001C  7A43 00B4                       LDAL     180.           'AT 180 OR INPT(0)
 001D  0660 0000     3    .INST        DO
 001E  B500 0006          .INST        SETUP,   LOOP           'PESET LOOP COUNTER TO PERFORM
 001F  7A43 0037                       LDAL     255.-200.      '200 MULTIPLY AND ACCUMULATES
 0020  0660 0000          .INST        DO
                          ;
 0021  0A20 0000  WAIT$   .INST        XTPR                    'READ THE EXTENDED PRODUCT
 0022  2020 0021                       JBR      0,WAIT         'IS READY BIT SET? CONTINUE LOOPING IF NO
                          ;
                          ;
 0023  7B43 000F                       ANDL     15.            'IF READY, SAVE ONLY 4 LOWER BITS
 0024  4C0B 0000                       SR1                     'SHIFT RIGHT ONCE TO GET RID OF READY BIT
 0025  FF00 0007                       STA      XTP            'STORE IN MEMORY LOCATION XTP
                          ;
                     4    ;
 0026  0920 0000          .INST        MSPR                    'IF READY,READ THE MOST SIGNIFICANT PRODUCT
 0027  FF00 0008                       STA      MSP            'STORE IN MEMORY LOCATION MSP
                          ;
                          ;
 0028  0820 0000          .INST        LSPR                    'IF READY, READ THE LEAST SIGNIFICANT PRODUCT
 0029  FF00 0009                       STA      LSP            'STORE IN MEMORY LOCATION LSP
                          ;
                          ;
                          ;
                          'DATA MEMORY STARTS HERE..........
       000000000          .DATA   0
 0000       0001  XADR$   1                                    'INSTRUCTION TO LOAD X BUFFER ADDRESS
 0001       0002  YADR$   2                                    'INSTRUCTION TO LOAD Y BUFFER ADDRESS
 0002       0003  ADR$    3                                    'LOAD IDENTICAL ADDRESS IN BUFX AND BUFY
 0003       0004  LDX$    4                                    'LOAD DATA INTO BUFX
 0004       0005  LDY$    5                                    'LOAD DATA INTO BUFY
 0005       0007  TASK$   7                                    'SELECT FUNCTIONS
 0006       0009  LOOP$   9.                                   'SET LOOP COUNTER AND START MULTIPLIER
 0007       0000  XTP$    0                                    'LOCATION TO STORE EXTENDED PRODUCT
 0008       0000  MSP$    0                                    'LOCATION TO STORE THE MOST SIG. PRODUCT
 0009       0000  LSP$    0                                    'LOCATION TO STORE THE LEAST SIG. PRODUCT
 000A  00000000C7 MEMR$   .BLOCK 199.                          'FILTER MEMORIES X(-199),X(-198),...X(-1)
 00D1  00000000B4 INPT$   .BLOCK 180.                          'INPUT BUFFER  X(0) X(1),... X(179)
 0185  00000000C8 H$      .BLOCK 200.                          'FILTER COEFFICIENTS  H(199),H(198),....,H(0)
                          ;
```

FIGURE A-2 (CONT'D)

LME